April 2009

Forerunner

Christopher Ivory
Worcester Polytechnic Institute

Dickson Ridley McCannell
Worcester Polytechnic Institute

Project Number: CR1-0801

Forerunner Project

A Major Qualifying Report

Submitted to the Faculty of the

Worcester Polytechnic Institute

In partial fulfillment of the requirements for

The *Degree of Bachelor of Science*

in *Interactive Media and Game Development*

and *Computer Science*

Dickson McCannell

Christopher Ivory

Charles Rich, Major Advisor

Dean O'Donnell, Major Advisor

Abstract

Forerunner is a mystery adventure game set in a steampunk science fiction world, developed to explore approaches to dynamic story generation and replayability. *Forerunner* uses a narrative engine specifically designed for this project that combines dialogue and action in a single story tree.

Table of Contents

Acknowledgements

We would like to thank Matthew Ivory and Elisabeth Beinke for their artwork in *Forerunner*. Both worked on the art as Independent Study Projects during B Term of the 2008-2009 WPI academic year.

Additionally, we want to thank the developers and forums of the Golden T Game Engine. The engine itself served as an excellent groundwork for the *Forerunner* game and the forums were an excellent resource which allowed us to avoid many common issues new developers have with the game engine.

Lastly, our sincere thanks go out to Professors Dean O'Donnell and Charles Rich. The two invested countless hours of time overseeing this project and offering us plenty of sound advice. We hope that the two found the experience as fulfilling as we did.

1 Introduction

Our main goal for this MQP was to explore the possibility of generating interactive stories. We see an interactive story as one in which the player would have more amount of freedom in comparison to the majority of games that are in the market today. In an interactive story game, the player would not easily be able to reproduce the exact same story, since the story would never unfold the exact same way twice.

After looking at various methods others had used to try to solve this problem in the past, we concluded that giving a player complete freedom in a game was not within our current capabilities, given the time available. We therefore decided to make a game that would explore a method of dynamic story generation.

This led us to design and create a novel type of narrative engine which controls the story flow of a game. The narrative engine contains every story element necessary to play through the game. The traditional game code we made controlled the graphics and interactions between the player and the environment, but the narrative engine did essentially everything else. The narrative engine itself would choose which path the story would take, which non-player character was the villain, and when certain scenes would take place within the game. The narrative engine also managed all of the game dialogue, as well as all of the logic for progressing through scenes in the game.

While *Forerunner* generates a lot of the major plot choices at runtime, the player can still affect the story as well. The player's actions throughout the game affect how much the NPCs trust the player character, which ultimately determines how difficult it is to prevent the final, fatal act of sabotage at the end of the game. The player must also correctly reveal the saboteur or suffer dire consequences. Other, pivotal choices are made by the player throughout the game which affects how events unfold.

Our game is called *Forerunner*. The game plot concerns of six passengers (one of whom is the player) stranded aboard a crippled spaceship, with another one of the passengers being dynamically selected by our narrative engine to be the villain. The player's goal is to discover who the villain is by noticing clues throughout the game that eventually point to the character's identity and confront him or her at the climax.

2 Background

Following are discussions of various sources we explored during the research phase of this project which were referenced in the design of *Forerunner*. There is a wealth of information on this topic, but we will discuss the most pertinent works which we drew upon while working on this project.

2.1 Literary

These are relevant books looked at during the research phase of the project that helped to define the early stages of the project.

2.1.1 Chris Crawford on Interactive Storytelling [1]

In this book, Crawford begins by creating one of the best definitions of interactive storytelling that I've ever seen. He systematically dissects several other definitions before presenting the reader with his own simplified version to reinforce his own definition, which defines interactivity as "a cyclic process between two or more active agents in which each agent alternately listens, thinks and speaks" (Crawford 29). From there, he moves one to describe many pitfalls with today's approach to interactive storytelling and how to change it. Crawford explains that by defining a story through levels of abstraction, the developer can still see the bigger picture while finding leeway in which to allow players freer reign and better control over the story's direction.

[1] Chris Crawford, 2004

Crawford outlines the steps he feels are necessary for creating a piece of innovative interactive storytelling technology. While the discussion is largely hypothetical and lacks technical examples, he provides reasonable arguments to support his ideas. He also focuses on characters' responses to drama and anticipation. He argues that it's not enough that drama is measured, but also that characters should interact to one another in logical ways. The author also delves into drama managers, which would observe the state of the game and intervene to heighten drama as necessary, either by including drama, shifting personalities or affecting the plot by other means. He reminds programmers that the goal of this would be to heighten drama as necessary, not necessarily to simply foil the player.

Chris Crawford's book is very insightful and resourceful. He provides many suggestions for improving interactive storytelling and backs those theories with reasonable arguments. Equally important however are his initial chapters on defining interactivity and storytelling for the readers. Crawford's definitions and ideas provided a solid groundwork for the *Forerunner* project.

2.1.2 Interactive Storytelling[2]

In his book, Glassner goes through and outlines all the major issues, as well as the myriad of possibilities, that arise when true interactive storytelling becomes a reality. Glassner's many examples from popular media help to illustrate his points.

Especially interesting was Glassner's discussion of what he called "programmability and adaptability." He argues, "The key to creating a new form of participatory fiction is the creation of systems that are programmable and adaptive." According to him, the greatest use of programmability is that it enables us to create more complex and realistic simulated environments, which in turn can capture the attention of the players and draw them in the way movies and plays have been creating sets for a long time.

[2] Andrew Glassner, 2004

The major downfall of Interactive Storytelling seems to be the fact that Glassner spends extremely little time discussing the actual applications of the concepts he outlines. He instead devotes the book to detailing the problems to be faced with the goal of creating an interactive story, but doesn't often touch on how to actually solve those problems. Nevertheless, the problems he poses are still very well thought out and argued in his book, and seem to provide a good basis for what to look out for when creating an interactive story.

2.2 Games and Tools

Below we discuss are various pieces of software found to be relevant to the project. Both games and narrative tools were explored in an effort to gain a better understanding on previous work on interactive story generation.

2.2.1 Façade

Facade was a very immersive and somewhat interactive story simulation. Though the immersiveness can be broken, the game handles a surprising number of player choices that would seem plausible within the confines of the game. The most fascinating part with relation to this project was the way in which the lesser details could be altered by the player while the AI still led the story to a common end. The story summary is nearly identical regardless of what route the player takes.

While players in *Facade*can affect the story, the game ends always ends with the player being shown the door. Conceivably, the entire game story in Façade could be expanded upon so that the scene in the apartment could be just one moment in a larger story. The designers gave the players enough control to alter the outcomes of the scene so that the end of the story can change while still preserving some authorial intent such as where the scene begins and ends.

Facade is a very innovative game which takes a unique approach to dynamic story generation. Though there are limitations in the open-ended approach to player responses, the game functions well

assuming the player reacts within the realm of plausibility. *Facade* succeeds as a working demonstration for greater immersiveness and a dynamic story-world.

2.2.2 *Wide Ruled*

Wide Ruled is a mechanism for generating stories while preserving authorial intent. It uses characters, settings, and plot points produced by an author to generate a complete story. The user is even able to define character relationships and roles to ensure that the story generation is not completely random.

One problem with *Wide Ruled* is that it does not currently allow for any complex player interaction; the most an author can do is offer the player a choice at a certain juncture in the story. The main problem, however, is the fact that *Wide Ruled* does not allow for much development of characters or plot, since the characters and settings are randomly determined by all possible values that could be each time the story is run. The plot points themselves are completely written out by the story's author and only vary in which characters or settings they involve and which order they take place in.

3 Story

Forerunner takes place in a future world with a steampunk setting, which we felt would set it apart from other games, since that is not a commonly used genre. The main plot of the game unravels on board a spaceship en route to a colony on the planet Mars. After a series of especially violent protests on Earth prior to the story, weapons of any sort were banned on the colony, and it became a place where people could go to live in peace. The ship, called the Forerunner, is carrying six passengers to the colony on Mars, along with a secret cargo hidden away in a sealed room. The passengers are the main characters of the story.

We made the decision to use a small group of characters because it allowed us to give the characters we did have more depth. The decision also made the characters easier to manage in the story tree, since the number of branches at certain places in the tree grows very quickly the more characters there are. The small area on the ship the player is restricted to prevented us from having to draw an entire world in the game.

The game's protagonist is named Neil Connors. As the player character, Neil does not have a full backstory, since that could make the player feel as though he or she is watching another character instead of himself or herself. We did choose to give the player the name of Neil, although if we revisited this project again we would like to give the player the opportunity to choose his or her own name.

The game opens with Neil, the player, waking up after a large explosion shakes the Forerunner. We tell the player right away in a status message on the screen about the explosion, but nothing else, so the player will become curious about the event and explore the ship to discover what happened. As the player explores, he or she discovers that the entire crew of the ship is dead, and the way to the ship's bridge is blocked off with rubble. The only survivors are the player and the five other passengers; this was done to evoke a sense of panic in the characters and the player, since no one left on board has any idea how to work the ship or save themselves.

Once the player has spoken to all five of the other passengers, they all hear another explosion and rush to the source of the sound, the engine room. We wanted to ensure that the player had been introduced to all of the characters in the game before we moved the story forward, and since the other passengers are scattered throughout the walkable section of the ship, the player must also become familiar with the area the game the takes place in before a lot starts happening within the game. In the engine room, they find that the backup communications console has been destroyed, so that they have no way of sending a message for help. The fact that the destruction was so targeted reveals that the

damage was sabotage, and since the passengers are the only people remaining on board the ship, one of the must be the saboteur. We decided to use sabotage as the catalyst for the story so we would have a villain hidden among the passengers that could conceivably be any of them; this allowed us the freedom to dynamically choose the villain.

The player is given the opportunity to converse with each of the other passengers and discover their backgrounds, as well as their possible motivations for sabotage. Eventually, the player must come across a keycard that unlocks the sealed door to the ship's cargo hold. The keycard acts as the transition to the next act; we used a different method to transition between each act. Once the room is opened, the characters converge to open a crate that was stored in that room. The passengers are shocked to reveal a large supply of weapons in the crate, despite the myriad of precautions each of them had to go through before boarding to ensure they were not smuggling any weapons to the Mars colony. We decided to use illegal weapons as a motive for the sabotage because it actually allowed us a variety of motives. For instance, one of the characters wants to steal the weapons for a group he is a part of to use, while another character wants the weapons destroyed and is willing to sacrifice herself and everyone else on board to prevent them from reaching the colony.

After a set amount of time passes, the saboteur strikes again, this time by murdering one of the other passengers, whether intentionally or accidentally. We used the method of time passing to transition to the next act here to show the capability of timers we had built into the story tree. The death had multiple reasons behind it. First, a dramatic event as jarring as a character's death would help keep the player's interest in the game. Also, the character who dies is obviously not the saboteur, so the player is able to eliminate a suspect. The other passengers are angered and resolve to take care of the saboteur before doing anything else; this allowed us to progress to the scene where the player must accuse one of the passengers of being the saboteur. The resulting confrontation reveals the character's

motives and allows the player to talk the villain into letting the others go free. We decided a dialogue-based confrontation scene would work out best for us since our game as a whole was mostly dialogue-based. Using the same game actions we had used throughout the game let the player continue doing what he or she had become familiar with.

4 Challenges & Solutions

There were many hurdles to overcome during the development of *Forerunner*. This section outlines some of these.

4.1 Dynamic Plot Point Generation

Dynamically creating the plot of a story was one of the major goals of the project. This section discusses how that goal was approached.

4.1.1 Avoiding Predictability

The majority of games do not vary much in story if someone plays them multiple times. Each game contains the same beginning, the same major plot points, and the same conclusion on each play-through. Often, details may change depending on player actions, but the player can expect the same basic story over and over again.

One major goal of this project was to discover a method of avoiding this predictability and allowing the player to gain a new experience each time he or she plays through the game. Doing so increases the game's replayability, since it becomes more unlikely the player will get bored with the game after a single play-through, and effectively increases the number of hours of enjoyment a player can take from the game overall. A major issue we faced while avoiding predictability was determining how to ensure that players would still be able to reveal the saboteur using logic. We did not want to

decrease predictability by forcing players to make random guess. In the following section, we explain the solution to this problem.

Another method of ensuring the player's experience remained entertaining was to continuously present the player with meaningful decisions. If every choice the player makes in the game has some impact on something later on, the player will be left wondering what would have happened had he or she picked the other choice, and will be more likely to replay the game to find out. This also helps make the game less predictable, since the player will never know how making a single different choice will significantly alter any interactions that happen for the rest of the game. In *Forerunner*, each choice has some meaning; even the different ways the player can respond to comments made by the non-player characters can at the least influence the trust the character has for the player and will make the final confrontation scene easier or more difficult depending on what that trust value has become.

With each play through the game, *Forerunner*'s narrative engine selects different values for many of the plot points, ensuring that the player's experience varies. Many of these choices by the narrative engine are influenced by player decisions or earlier narrative engine choices. For instance, the character who the narrative engine picks to be the saboteur cannot be in conversation with the player at the time the saboteur is chosen, since that is when the first sabotage that takes place within game happens. Likewise, the character who dies cannot be the saboteur, as that would prevent the story from reaching a climax.

The climax is an important part of the way *Forerunner* avoids predictability. While every game will end with the player accusing a character of being the saboteur, as long as the player is correct in his or her accusation, a different climatic scene will unfold depending upon the identity of the saboteur. Then, just as the player is given the opportunity to succeed, he or she is also given the opportunity to fail; the player will then view either a good or a bad ending. This means that a player who fails might

want to replay the game to try to succeed, or a player who fails might want to see what happens if he or she fails. Either way, the scene with the confrontation between the player and the saboteur itself will be different as long as the narrative engine randomly selects a different saboteur during the succeeding play-throughs. Also, each of the choices the player makes while talking to the saboteur throughout the game will impact the difficulty level of the final confrontation.

4.1.2 Revealing the Saboteur

Near the beginning of the game *Forerunner*, the narrative engine randomly chooses one of the non-player characters in the game to be the saboteur for that play-through. There are various hints scattered throughout the game that allow the player to narrow down the list of suspects until only one is remaining.

The first hint is revealed to the player. The saboteur is chosen by the narrative engine at the time the first in-game sabotage, the destruction of the backup communications console, happens. The character that the player is talking to at the moment of the sabotage cannot be the saboteur, which he or she points out at the end of the first group scene, since both that character and the player were in each other's company and so could not have caused the destruction of the console. This hint also suggests to the player that he or she must look for ways to eliminate other characters from suspicion, and so this first clue teaches the player what to expect in the future of the game.

The second hint comes from the player asking around for the keycard to open the sealed door to the cargo hold. Seeing the ship's cargo reveals the saboteur's motive, so the saboteur will never be the character to give the player the keycard. Like with the first hint, if the player is talking to a character when the death happens, that character cannot be the saboteur. The final hint is that the saboteur could not be the character who dies.

At this point, if the player has paid attention to all of the hints, he or she should be able to figure out who the saboteur is. If the player missed a single hint, he or she can still reach the final confrontation scene, since the game allows one false accusation. If the player missed multiple hints, and has only narrowed the list of suspects down to three or four characters, the only way to reach a good ending of the game is to guess the saboteur correctly.

4.2 Narrative Engine Design

The narrative engine that was created for this project and was used to create story of the game *Forerunner* is described in this section.

4.2.1 Alternative Approaches

Our goal in creating the narrative engine was to make a tool that could be used to flexibly author a story. The narrative engine would provide a relatively simple method for making a story that was dynamic and replayable. The actual variation within the story and the replayability of the game would be dependent on the author, but the capability would be there to make as dynamic and replayable as possible.

Early in the design process, we decided that non-player characters would have their own goals, emotions, and relationships with other characters. These traits, along with the characters' own preprogrammed personalities, would determine each of the characters' responses to any external stimuli within the game. These responses would take the shape of dialogue or some other form of action, such as walking towards a different area of the game world. The complexity of having an artifical intelligence directing the overall story and also influencing each of the non-player characters who would, in turn, have their own artificial intelligences and goals made this idea an unreasonable one.

One of the first approaches we looked at for creating a narrative engine was a model that would continuously loop, picking a scene in the game to generate each time based on various preconditions.

For instance, some scenes would not be able to be run until certain other scenes that they depended on had been. Once the scene had been picked, the scene would unfold in a series of “beats”, each one run after checking preconditions. Once a beat was chosen to be run, if its preconditions were met, it would perform its changes to the game world, such as setting a variable, printing some dialogue, or moving a non-player character. At certain times, the beat would wait for a player response before carrying out its actions. These player responses might include, for example, clicking on an object or choosing a dialogue from a number of options that were provided.

We also looked closely at Dynamic Experience Management in Virtual Worlds for Entertainment, Education, and Training. This approach looked at the flow of a story as different plot points connected in any way the story could possibly flow from beginning to end, with some plot points having the potential to be skipped during ny given run through the story. This story flow contained islands that indicated which plot points had to have happened before others could happen. Any path through the story that did not pass through all of the plot points connected to the islands was pruned and could not be taken. In this way, authorial intent was preserved in any incarnation of the story.

We decided to make all of the game events the same, so that dialogue could be run the same way as actions that change the game state. An idea looked at was to also use dialogue templates to allow for different ways for an idea to be conveyed; for instance, if a character who was supposed to say something died, another character would say something similar, but in that character's own style of talking. The final narrative engine does not utilize this idea but instead actually has multiple possible dialogue strings that can be spoken and chooses one of them dynamically based on certain preconditions.

4.2.2 Final Design

Eventually, a final design for the narrative engine was reached, and the engine itself was built. This section outlines the techniques used in its creation.

4.2.2.1 Abstract Design

In the final design of the narrative engine, an and-or tree represents all of the possible actions that can happen within the game. An and-or tree is a standard computer science method of representing choices, and is used to break down goals into subgoals, so it can be used to present the player with options as well as guide the player, non-player characters, and overall story.

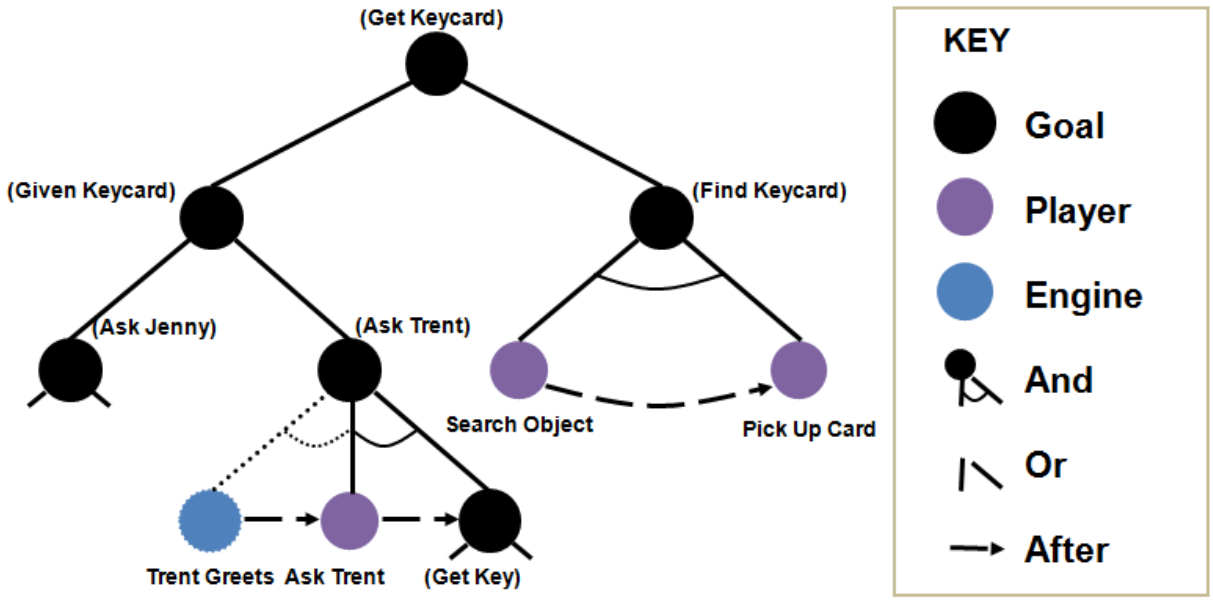

Figure 1: An abstract example of a section of *Forerunner*'s And-Or tree

In the example tree in Figure 1, an and node represents a goal that is only satisfied if all of its child nodes are satisfied; i.e. it must have all of its subgoals completed before it is considered complete. An or node represents a goal that only needs to have a single one of its child nodes satisfied; i.e. it has alternative methods to complete it. This goal could be a player mission, such as getting a keycard to open a door either by receiving it from a non-player character or finding it lying on the ground, or it

could be something hidden from the character, such as multiple possible dialogue strings that could be spoken by a single non-player character depending on certain events in the game. The leaf nodes of the tree each contain the actual actions that either print dialogue or otherwise affect the game state in some way. The leaf nodes also contain preconditions; the nodes will not be processed unless these conditions are met. Also, each node in the tree contains a priority; only the highest priority children of any node will ever be processed. This prevents every possible event in the game happening at once in a random order.

4.2.2.2 Technical Design

As with the game itself, the narrative engine was developed in Java. A method creates the tree out of *and*, *or*, and *leaf* node objects, each of which contains unique identifier, a priority, and a list of child nodes. *Leaf* nodes have no children and contain a list of preconditions which must be satisfied before they can be processed and a list of actions that are carried out when the node is processed. A *leaf* node is marked as visited when it begins to be processed, an *and* node is marked as visited as long as all of its children are, and an *or* node is marked as visited as long is at least one of its children is marked as visited. A node and it children cannot be processed as long as the node is marked as visited. One of the leaf node game actions that is used throughout *Forerunner* is one that marks a node with a given identifier as either visited or not visited; this can be used to ensure that nodes can be processed multiple times.

The Java code reads an entire and-or tree from a series of XML files. The XML contains and, or, and leaf elements that represents and, or, and leaf objects respectively that will be constructed from them. The XML also contains a number of Tree elements, from which the Java code reads in a tree from another corresponding XML file and then puts that tree's root node where the Tree element would have been. Node ID values and priorities are represented by XML attributes of their corresponding elements.

Preconditions and game actions are represented in the XML by text of the leaf elements. The Java code reads them in and translates them into Precondition or Action objects that are contained and used by the leaf nodes. The preconditions and actions are simple strings in the XML. Each leaf node can contain any number of actions and preconditions. There are also special not and or preconditions which contain other preconditions and enact certain rules when the Java code checks to see if they are satisfied or not.

4.3 Other Issues

This section describes other major problems that were encountered and solved during the creation of *Forerunner*.

4.3.1 Room Occlusion

In a mystery adventure game such as *Forerunner*, it is vital that the player not be omnipotent. Otherwise, vital plot details about the villain's whereabouts, motives and actions might be revealed too easily and the gaming experience would suffer for it. In order to prevent this from happening, steps must be taken to ensure that players are only privy to the information which is intended for the player character to see and know.

In *Forerunner*, it became necessary for players to only be able to see the events occurring in the room the player character is currently occupying. That way, the game could keep up the illusion that a saboteur is moving about the ship. For example, the sabotage of the engine room is scripted so that it cannot occur while the player is within that room, but it is conceivable that the player may be in a room nearby when the explosion occurs. If the engine room were visible to the player when he or she is within such a room, it would become apparent to the player that the scripted explosion occurs even though no NPC physically placed a bomb. By occluding the engine room from visibility while the player character is within another room, the illusion of a bomb being placed is enough to suspend disbelief.

Figure 2: The entire *Forerunner* floor plan

In games with a first-person view, room occlusion is done automatically. Most games do not allow a player to see through walls, so the actions taking place in adjacent rooms are hidden from the player. In *Forerunner*, a 2D isometric view is employed. This means that the entire map is viewed from the third person, and as such adjacent rooms are not hidden from the player by default and must therefore be handled manually by the game.

The map of the ship in *Forerunner* is non-uniform and does not lend itself to any form of grid. Therefore the process of determining which room the player is in is not trivial. The game engine solved this by collision detection. An invisible, one-pixel image is located at the center of the base of the player's character while each room of the ship is mapped with its own invisible base. Each update cycle, the marker image on the player base does a collision check with the room bases. The room base colliding player marker is the room in which the player's character is standing. The rooms of the ship are divided as shown in the figure below.

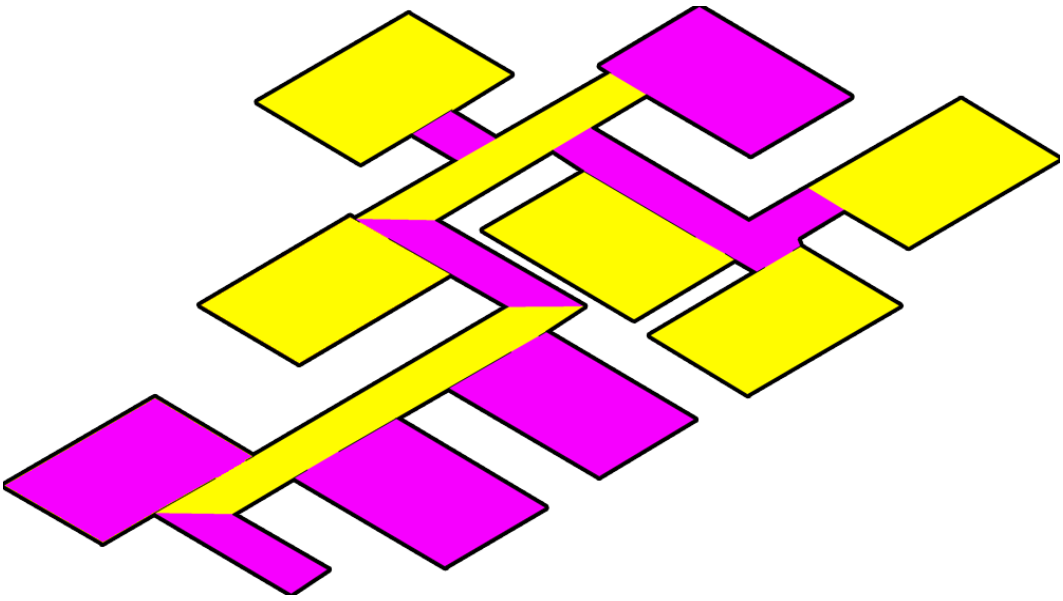

Figure 3: Map separated by rooms

Once the room the player is standing in is discovered, only the details, NPCs and objects within that room are rendered during that update cycle. All other rooms and hallways are not drawn and therefore not visible to the player. In this way, only events and objects occurring within the same room as the player character are visible. All other actions are hidden, thereby effectively keeping the player from learning more about the story than his character knows.

4.3.2 Dialogue Display

Conversation is the most important way in which a player can interact with the *Forerunner* world. Gathering clues from talking with the NPCs is one of the primary means to determine which character sabotaged the ship. The game also has several group dialogue scenes which, if displayed poorly, could become difficult to read and hinder the player's ability to enjoy the game. Given this, it is incredibly important that the players be able to read the conversations in a user-friendly way.

One of the big issues with dialogue display stemmed from the fact that in group scenes with five or six characters, it became quite possible that the amount of dialogue provided by the NPCs between the player choices would be too large to fit on the screen all at once. Dumping all of the dialogue onto

the screen would not only be a cumbersome amount of text for the player, but early chunks of dialogue would prematurely scroll off the screen before the player had a chance to read it. To resolve this, *Forerunner* was designed so that pauses could be added after any line of dialogue which would require user input to continue. This allowed for long conversations to be broken into readable chunks in such a way that players could control the flow of conversation to suit their preference. In this way, players would not be overwhelmed during the dialogue in group scenes and no parts of the conversation would be missed.

Another major issue stemmed from the group dialogue scenes. In situations where five or six characters might all be talking at once, there is a great potential for an individual character's mood and motive to become unclear. In an adventure game such as *Forerunner*, assessing a character's actions and intentions are important in determining which character might be the villain. The dialogue must be displayed in such a way that individual characters can be identified quickly and styled so that their parts of the conversation can easily be analyzed on independently.

Figure 4: Color-coded dialogue in a group scene.

Forerunner solves these problems by color-coding and placing the dialogue so that lines spoken by character match the hue and alignment of the character's portrait (see Figure 4). For example, Jenny's portrait and dialogue are both in green. When her portrait is placed on the left side of the screen, lines of Jenny's dialogue will be green and aligned to the left side of the screen. Each character has its own color and character portraits are alternately placed on the left and right sides of the screen when the conversation starts. As a result, all of the lines of dialogue can easily be associated with the corresponding character at a quick glance, thereby making group scenes very easy for the player to process.

5 Conclusion

5.1 What Worked

As with any game project, not everything goes according to the original plan. However, there are several parts of *Forerunner* that we feel worked out quite well. These aspects of the game worked as expected or were parts we discovered along the way which we were particularly proud of.

5.1.1 Narrative Engine

The primary technical experiment of the *Forerunner* project was the narrative engine. It drives the story forward and generates plot points dynamically each time a new game is started. In this sense, we feel the narrative engine is a success.

Each time a new game is started, information such as who the saboteur is, who finds the keycard and who dies is generated by the game. This makes it highly unlikely that players will have identical gaming experiences even if they make the same choices along the way. It makes the game much more dynamic and increases the potential for replayability.

Additionally, the narrative engine was designed in a somewhat unorthodox manner in that all of the actions taken by the game, including dialogue, are stored in a single tree. Traditionally, games store dialogue in separate trees. In *Forerunner*, having dialogue occur within the same tree as all other actions allowed us to more easily use dialogue within our AI structure. Conversations could be used as preconditions to events and events were able to occur while dialogue was taking place, such as passing items to the player, moving NPCs and so forth.

5.1.2 Game Style

We initially chose an adventure style for this game rather than an RPG or FPS because we did not want to burden ourselves with gameplay mechanics such as weapon balancing, combat AI other

issues more indicative of combat games. By making *Forerunner* an adventure game, we were able to invest the time which would have been spent into those mechanics and focus on the game story itself and the narrative engine we were designing. While in theory our design for a narrative engine could work in other game genres, implementing the narrative engine at first in an adventure game (which is traditionally associated with dialogue and puzzles over shooting and action) allowed us to focus on the aspects of the game we felt were most important.

5.1.3 Room Occlusion

As mentioned previously, one big issue in game design was ensuring that players were not privy to information that should be inaccessible to his or character. Otherwise, it would be quite easy for a player to get the upper hand using knowledge that should not be known. It therefore became necessary to occlude parts of the map not visible to the character from his current position.

Collision detection was used to check which room the player character was in, then all other rooms and corresponding entities were removed from the render cycle. This very effectively blinded the player to the events occurring in other rooms. Not only did this make it easier for the saboteur to act in relative anonymity, but it also allowed us to design scenarios in which we could create the illusion of a character sabotaging the ship without having to physically move the saboteur to commit the crime. Essentially, we were able to save programming time by no longer having to script actions which the player shouldn't be allowed to see by the narrative engine.

5.1.4 The Plot

The narrative engine was one of the core components of our project, and as such it became very important that we come up with a story which would properly highlight the replayable nature of the narrative engine. All the while, we still had to include enough structure to show that the engine could

also preserve authorial intent. We feel that the plot of *Forerunner* successfully balanced the combination of dynamic plot point generation and preservation of authorial intent.

Regardless of the variables in *Forerunner*, the general plot remains the same with repeated play: a saboteur cripples the ship, the weapons are revealed and a character is murdered. A successfully completed story always culminates in a confrontation between the saboteur and the player character, in which the player must calm down the saboteur. While *Forerunner* has several dynamically generated plot points, the story can be generally summed up the same way each time because it preserves the authorial intent.

5.2 What Didn't

While there were several aspects of the *Forerunner* project we were proud of, there were also some aspects of the game that did not live up to our expectations. We would like to share these cases in the hopes to help others avoid the mistakes we made. The experiences have each taught us something, allowing us to avoid such hurdles in future projects.

5.2.1 Team Communication

On the whole, team communication for this project worked quite well. The game engine integrated with the narrative engine with minimal effort and most parts of the project merged together as we intended. However there were some scenarios where we felt a more constant flow of comments would have been beneficial.

Forerunner has several different art assets which appear to stand out from the rest of the game. For example, the character models clearly have a different stylistic feel from the background and other objects in the room. This is the result of a lack of communication between the artists during the development phase. Also, several art assets needed where not completed by the end of the artists' time on the project. This was due to either a lack of formal documentation requesting these assets or our

own inability to foresee the need of such artwork. In either case, it is clear that better and more constant communication with and between the artists would have yielded more ideal results and a more uniform art style in the game.

5.2.2 Scripting Language

We opted to design our own scripting language for this project for several reasons. By implementing a scripting language of our own design, it would have the functionality we needed and could be expanded upon to suit the needs of the project. The language could also be designed to work intuitively with the narrative engine in such a way so that the scripting language and narrative engine could be managed in other projects with minimal effort.

However, designing such a scripting language from the ground up took a lot of time and effort. The language itself worked well in *Forerunner* and with additional coding could be expanded upon, however it is initially much more restricted than other scripting languages. JavaScript and Lua both have more native functionality and would have made narrative language more robust and less time consuming. Also, a more common scripting language would reduce the learning curve for programmers wishing to make use of the narrative engine.

5.2.3 Game Engine

The Golden T Game Engine (GTGE) is a very well-designed Java game engine. Initially, GTGE seemed like a perfect fit for this project. However as *Forerunner* got more complex, it began to push the limits of what the engine was designed to handle. For example, GTGE input listeners were designed to process one key at a time as well as only basic mouse functions. As our user interface called for more detailed mouse-work , we had to work around issues that arose in the GTGE. Given the opportunity, we would look at other Java game engines (such as the jMonkey Engine) that might be better suited for a game such as *Forerunner*.

5.2.4 Team Size

During the course of this game, it became very apparent that the project would have benefitted from a third team member with an artistic background. As one of us worked on the narrative engine and scripting language, the other worked on game design, the user interface and input. Additionally we both worked on the script and added to the art assets that hadn't been completed by the artists.

In a game such as *Forerunner*, the script of the story is just as important as the underlying technology. Without a proper story to demonstrate the power of the narrative engine, there is no reason to design it. While *Forerunner* has a compelling narrative, we feel if we had a third team member to be a dedicated script writer that the game plot would have been more robust and defined. Also, the artwork could have used more polish and several more art assets could have been used. Having a team member with the skills necessary to make such changes would have been very helpful.

5.2.5 Script Editor

The narrative engine of *Forerunner* loads the game tree from XML files to determine all of the game's preconditions, actions and dialogue. While storing the data in this fashion works quite well, it makes it difficult to quickly edit and expand scenes of the game. Given the opportunity, we would have liked to develop an XML editor applet designed specifically to display the XML files loaded by the narrative engine. This would allow programmers to design, read and modify game scenes much more efficiently and effectively. We initially discarded this idea as an inefficient use of our time. However, in retrospect we feel it would have saved hours of time scripting the game scenes.

5.3 Project Summary

The *Forerunner* project allowed us to explore a new approach to an old gaming problem. We were given the opportunity to research the issue of dynamic story generation, examine previous and current approaches to the problem and implement a solution of our own making. Additionally, we

designed a game to test that solution. Over the course of a year, we were able to take this project from infancy to completion.

In *Forerunner*, we see both aspects of the game we are proud of and parts of the game that we would approach differently the second time around. While we did not achieve all of the goals we set out to, both our successes and failures were turned into learning experiences which will help us as we move on to future projects.

Bibliography

Mat Buckland, Programming Game AI by Example, Wordware, 2005.

Crawford, Chris. Chris Crawford on Interactive Storytelling. Grand Rapids: New Riders, 2004.

Glassner, Andrew. Interactive Storytelling : Techniques for 21st Century Fiction. London: A K Peters, Limited, 2004.

Riedl, Mark O., Andrew Stern, Don M. Dini, and Jason M. Alderman. "Dynamic Experience Management in Virtual Worlds for Entertainment, Education, and Training." International Transactions on Systems Science and Applications Copyright (2008).

Salen, Katie, and Eric Zimmerman. "Games as Narrative Play." Rules of Play : Game Design Fundamentals. New York: MIT P, 2003.

Skorupski, James, Lakshmi Jayapalan, Sheena Marquez, and Michael Mateas. "*Wide Ruled*: A Friendly Interface to Author-Goal Based Story Generation." (2007). University of California, Santa Cruz.

Appendix A: Original Plot Pitches

Pitch 1

The player's character is aboard a spaceship in a science fiction story-world. [The ship has just recently picked up an ambassador of Earth from forming a tentative truce with an alien race - a decision that had been widely criticized from factions of both species. - (Really cliché basis for the plot, probably either come up with something better or just remove this and have the ship simply out in deep space or returning home from a mission.) As the crew attempt to return home, an explosion on board leaves the ship dead in space.

With only a few hours before the ship tears apart, the crew must make their way through fused bulkheads, malfunctioning security systems and other obstacles as they make a last-ditch effort to reach the escape pods. Only by working together do they have any hope of escaping, but when it's discovered that the explosion was the work of a saboteur, everyone becomes a suspect. Can the player figure out who amongst the crew is a saboteur and escape the ship in time?

Pitch 2

The player's character is exploring a cave when the entrance collapses behind him. In the section where the player is trapped lies a large egg. It soon hatches and from it pops an unusual and magical creature that seems to be growing and evolving at a rapid pace.

The creature grows when it's goals are met, and it's up to the player to interpret what it needs. Every growth spurt, the creature will grow in unexpected ways which will alter its mood and needs. If the creature isn't taken care of properly, it may get angry or hungry and decide to make the player its next meal! The player's goal is to allow it get big enough so that it can break out of the cave and free them both.

Appendix B: Design Documents

Introduction

Most interactive adventure games to date have relied upon pre-specified decision trees and scripted dialogue to generate a compelling narrative. While this allows for some well-designed stories, it severely limits the re-playability of those games.

What *Forerunner* attempts to accomplish is the dynamic generation of the main plotline of a game which is attuned to player's choices and the conditions of the game environment in such a way to maintain the qualities of a compelling narrative. The dynamic story generation will take into account concepts such as drama, the definition of a main character, and the notion of narrative flow. The game will focus on creating a re-playable, compelling interactive experience which will encourage players to explore the story world several times through.

Story Info

Summary

Forerunner focuses on the events that unfold on the *Forerunner*, a transport starship, in the hours after an explosion on the bridge cripples the ship. The player takes on the role of Neil Connors, a reporter who was given an anonymous tip that the flight was more than it appeared to be. With the crew dead and power failing, Neil and the other civilian passengers must work together to escape alive.

As events unfold, Neil begins to understand why this flight was so important. It becomes clear that the explosion was no accident, and that the saboteur has been on board since the start. With everyone as a suspect, Neil is forced to wonder: can anyone be trusted?

Plot Outline

The game opens with the player, Neil, waking up after an explosion rocks the ship. After exploring, Neil comes to the conclusion that the entire crew has been killed and the ship disabled. The only people left alive besides Neil himself are the other five passengers. Soon, there is another explosion, destroying the radio, simultaneously stranding the passengers with almost no hope of rescue and proving that one of the survivors is actually a saboteur.

Soon a weapons crate is discovered on the ship, despite weapons being illegal on the Mars colony to which the *Forerunner* was headed. It quickly becomes clear that the motivations of the saboteur, as well as the tip the reporter protagonist received, revolve around these smuggled weapons.

Neil has a choice: Either discover who the saboteur is and confront him or her, or simply ignore the very dangerous presence among the passengers and work to fix the ship in order to reach safety.

Regardless of Neil's choice, the story culminates in a climatic confrontation with the saboteur.

Setting

The entire game will take place aboard the transport ship, the *Forerunner*, in the near future (about 100 years ahead of present day). While playing *Forerunner*, the player will experience the game from the perspective of a civilian on board the ship in the hours after the primary power is lost. The ship is stuck midway between Earth and Mars.

Characters

Neil Connors (Player)

The player takes on the role of Neil Connors, a reporter. Neil boarded the transport following an anonymous tip that the *Forerunner* was transporting something for the government in secret. As Neil, the player must delve into the mystery that unfolds on the ship before it is too late.

Percy Hawkins

Dr. Persephone Hawkins, or Percy for those close to her, is an expert surgeon stationed on the Mars colony. On a professional level, she is the definition of success: sharp, strategizing and self-reliant. Socially, the young doctor is somewhat of an outcast. She'll rarely crack a smile at a joke and always dresses for business. She prefers male friendships to female ones, seeing them as more practical. However, for Percy, romance always takes a back seat to professionalism. The term "Ice Queen" has been tossed around more than once during the course of her career.

Percy was raised in a well-off, very structured household. Though her father was also a doctor, it was common knowledge that he smuggled medical supplies to those who need them - often through questionable means! From a young she was molded to be very proper, and very reserved. When she was 13, her father was shot by a couple of thugs who broke into their house for money. In her grief, and in his honor, she redoubled her efforts in her studies, and graduated a year early at the top of her class. She took a low-pay job on the Mars colony, and is returning after attending a medical conference on earth.

Rachel Wylde

Rachel is a small-time actress of some acclaim. Having been somewhat typecast as an action character, the young woman tries to live up to her roles by staying in shape and trying extreme sports and activities. When there's danger, she challenges herself to face it head on just as her characters do in movies. While this doesn't always work out as she intends, she continue to try and live up to her larger-than-life, fictional counterparts. Rachel is one of the civilians aboard the *Forerunner* on its voyage to the Mars colony, heading out a few days before a film shooting to take in the sights and try some null-atmo rock climbing.

Recently, the tabloids have relentlessly followed Rachel's waning career. Not a week goes by without another rumor of her love life or possible drug use. Most of popular opinion tends to see her as

a has-been, and don't take her seriously anymore. Though she claims to be heading to the colonies for a new movie, there's little evidence to back this up.

Steve Farro

An accountant for a minor corporation, Steve Farro is still far from what might be considered a stereotypical bookworm or "math guy". He's kind of the muscle man of the group trapped on the ship, but he retains an easy going personality and is not one to easily make enemies or hold grudges. Steve lives in the suburbs of small city with his wife, three children, and two dogs, and likes to spend his free time on the tennis court. Right now, he's on a pretty routine business trip to meet with some contacts from another company.

In the past few years, a few of the activists he's worked with have been shot while picketing. Ever the family man, Steve plans to take his family to the peaceful colonies as soon as he can afford to. As a result, he has a very personal devotion to the anti-gun laws on Mars and would go great lengths to make sure they were followed.

Jenny Porter

A quiet, though not unfriendly, young woman, Jenny is the type of person who doesn't like to take risks. This quality can be both a good one and a bad one, since she's the last person anyone would ever expect to get into trouble, yet she has never been very good at talking to people. She definitely thinks of the friends she does happen to make as the most important things in her life, and they all love her the same way, but her incredibly quiet nature has kept that number of people to a minimum.

Trent Porter

Trent is Jenny's brother. Like her, he may not be the most outgoing person ever, but he has always been the type to take charge in a bad situation, partially because he is never one to panic. He may not be the first one to volunteer to lead, but he's willing to step up to get the job done if no one

else does. Trent and Jenny were taking a trip together to see their mother, something they like to do every two years. They don't really get to see her otherwise, since she lives on a different planet than they do.

Technical Info

Key Features

Character Agents

Character interaction is a large part of *Forerunner*. The game focuses on making the NPC agents reactive to the world around them, each with their own personality and routine. The characters will move about the story world, fulfilling their routines until such a time where changes from the narrative engine or stimuli from player interactions supersede those tasks. In such cases, mix-in behaviors will be used to transition the change in priorities.

The narrative engine will handle dialogue which should be considered pivotal to the storyline. However, players can initiate dialogue with NPCs on their own to learn more about the characters and to alter relationships throughout the story.

Story Structure

The story will follow a path to a single, or possibly one of two, climatic scenes. The path will be determined by the player's actions as well as any objectives of the drama engine that have been met or need to be met. In other words, events will happen in the story if a certain objective in the story, such as one character being critically wounded by an explosion caused by the saboteur, needs to be met before other paths in the story will continue. Going along with that, any path in the story will never happen unless certain objectives are met to enable it. In the previous example, this might be the character walking into the room in which the saboteur had planted the explosive. If the character doesn't enter

the room, he or she can't be injured, so the explosion won't happen until the character is actually present.

Narrative Engine

The narrative engine is a drama manager that will be the driving force behind the progression of the story. As the player interacts with the story world, it will monitor conditions of the player, character agents, and objects. Using the story structure as a template, the narrative engine will alter the course story world attributes in the story to guide events dynamically. The narrative engine will predict the amount of drama events instill, and generate a story that gradually increases in drama until the story climax is reached.

The narrative engine is an and-or tree. Each parent node represents a goal which is broken down into sub goals and base actions which can be taken within the game. Its leaf nodes contain actions to be taken. These include dialogue spoken by either the player or an NPC and changes to the world state.

Controls

Forerunner will emphasize intuitive controls to allow for a minimal learning curve that does not detract from the game play experience. This will allow veteran and novice computer users to share in the same experience. The controls will be as follows.

- ***Esc Key:*** Access Menu
- ***Mouse Button 1:*** Move Avatar / Menu Selection; Interact with object or character

Required Art Components

The following assets are required to adequately populate *Forerunner* with artwork. Each of the following must be generated in a sprite-compatible format, such as GIF, JPG or PNG.

- 6 characters, each modeled and animated

- 6 character portraits
- 5 isometric floor tile sets
- 5 isometric wall sets
- 15 isometric objects
- 1 pop-up menu with 5 buttons
- 1 confirmation window with 2 buttons
- A bottom bar HUD to manage dialogue
- 2 splash screens

For more information on the artwork, go to the Artistic Design Document.

Engine

To achieve this particular style of game, the Golden T Graphics Engine (GTGE) will be used to process the graphics in a Java environment. Research has been done to assure that GTGE can adequately fulfill the graphical processing requirements of the game as well as potentially providing hardware acceleration for fast processing speeds.

GTGE can generate the objects within the story world as a series of sprites attached to a tileable, isometric background layer. The engine can handle the animation of sprites, interactive GUIs and so forth. It also allows for easy camera panning and response to user input.

Design Concerns

Run-time Processing - The graphic and drama engines need to be processing in real time so that the game maintains accept frame rates.

Drama Manager Dominance - The drama manager must affect the goals and actions of the character agents in such a way to sustain believability. Character agents must not act outside of their nature.

Artistic Info

Style

The game will be designed from a 2D isometric perspective. While isometric perspective adds some illusion of a 3D space, the restricted camera angle and lack of depth perception allow for the game to be generated in 2D sprites. Below is an example of an isometric structure:

Atmosphere

Given that the game will be serious in nature, with emphasis on character structure and dialogue, the game will have an element of realism and avoid an idealistic vision of the future. The ship will be dark and gritty in some places, with the glow of console light illuminating others. Stylistically, the game will mirror the theme of futuristic steampunk, margining class and technology.

Character Style

The characters in the game should have very formal, unexpected attire compared to that of traditional sci-fi games. Their clothing will be very similar to standard steampunk garb.

Environment Style

The ship itself will be designed to stylistically represent a merger between advanced technology and steampunk. Though some aspects of a Victorian, classy culture should be present, there should remain an element of grit and dirt from pollution of the engine. The ship should have been, at one point, an idealistic representation of class that has slowly degraded with time.

Level Design

The basic level design of *Forerunner* is comprised of one setting. All the events in the game will take place within the rooms of the ship. Because the game is designed to be dynamic and flowing, there will be only a single level in the game. The entire story world will be connected and accessible throughout the game.

Sound and Music

Though audio media is not one of the key functions in *Forerunner*, a background soundtrack and possible atmospheric sounds will be added to the game to add to its immersiveness and overall enjoyment level. Sounds will also be used where appropriate for explosions and other sound effects involved in the interaction with the story world.

Appendix C: Sample Story Tree

```
<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT Tree (#PCDATA)>
<!ELEMENT And (Tree*, And*, Or*, Leaf*)>
<!ELEMENT Or (Tree*, And*, Or*, Leaf*)>
<!ELEMENT Leaf (#PCDATA)>

<!ATTLIST And ID ID #REQUIRED>
<!ATTLIST And priority CDATA #REQUIRED>
<!ATTLIST Or ID ID #REQUIRED>
<!ATTLIST Or priority CDATA #REQUIRED>
<!ATTLIST Leaf ID ID #REQUIRED>
<!ATTLIST Leaf priority CDATA #REQUIRED>
<!ATTLIST Leaf playerNode CDATA #REQUIRED>


<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE And SYSTEM "forerunner.dtd" >
<And ID="XA1" priority="1">
	<And ID="a2" priority="1">
		<Tree>Death-Explosion.xml</Tree>

		</And>
<Leaf ID="l1" priority="1" playerNode="false">%DialogueFrame:true,0,1,2,3,4,5</Leaf><Leaf ID="l2" priority="1"
playerNode="false">DFrame:false%SetVisited:l1,false;SetVisited:l2,false</Leaf>
	<Leaf ID="idvalue0" priority="0"
playerNode="false">Not:Value:1001%SetRandomValue:81,82,83,84,85</Leaf></And>


<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE And SYSTEM "forerunner.dtd">
<And ID="XA1" priority="1">


	<And ID="XA8" priority="1">
<!--		<And ID="XA9" priority="1">
			<Leaf ID="XL7" priority="1" playerNode="false">%PutCharacter:0,0</Leaf>
			<Leaf ID="XL8" priority="1" playerNode="false">%PutCharacter:1,0</Leaf>
			<Leaf ID="XL9" priority="1" playerNode="false">%PutCharacter:2,0</Leaf>
			<Leaf ID="XL10" priority="1" playerNode="false">%PutCharacter:3,0</Leaf>
			<Leaf ID="XL11" priority="1" playerNode="false">%PutCharacter:4,0</Leaf>
			<Leaf ID="XL12" priority="1" playerNode="false">%PutCharacter:5,0</Leaf></And> -->
		<Leaf ID="XL1" priority="2" playerNode="false">%Status:Reporter Neil Conners wakes up as an explosion
rocks the ship Forerunner.;SetValue:1001;SetValue:14;SetValue:15;SetValue:16;SetValue:17</Leaf></And>


	<And ID="XA12" priority="2"><And ID="XA2" priority="1">
		<And ID="XA3" priority="1">
			<Tree>Trent Intro.xml</Tree>
			<Leaf ID="XL2" priority="1"
playerNode="false">Value:1;Value:1001%DialogueFrame:true,0,1</Leaf></And>
		<And ID="XA4" priority="1">
			<Tree>Jenny Intro.xml</Tree>
			<Leaf ID="XL3" priority="1"
playerNode="false">Value:2;Value:1001%DialogueFrame:true,0,2</Leaf></And>
```

<And ID="XA5" priority="1">
<Tree>Percy Intro.xml</Tree>
<Leaf ID="XL4" priority="1"
playerNode="false">Value:3;Value:1001%DialogueFrame:true,0,3</Leaf></And>
<And ID="XA6" priority="1">
<Tree>Rachel Intro.xml</Tree>
<Leaf ID="XL5" priority="1"
playerNode="false">Value:4;Value:1001%DialogueFrame:true,0,4</Leaf></And>
<And ID="XA7" priority="1">
<Tree>Steve Intro.xml</Tree>
<Leaf ID="XL6" priority="1"
playerNode="false">Value:5;Value:1001%DialogueFrame:true,0,5</Leaf></And>
<And ID="XA10" priority="1">
<Or ID="XO1" priority="2">
<Leaf ID="XL14" priority="1"
playerNode="false">CharInRoom:0,6;CharInRoom:2,7%SetRandomValue:102,103,104,105;SetValue:23</Leaf>
<Leaf ID="XL15" priority="1"
playerNode="false">CharInRoom:0,6;CharInRoom:2,6%SetRandomValue:103,104,105;SetValue:23</Leaf>
<Leaf ID="XL16" priority="1"
playerNode="false">CharInRoom:0,7%SetRandomValue:101,103,104,105;SetValue:24</Leaf>
<Leaf ID="XL17" priority="1"
playerNode="false">CharInRoom:0,11%SetRandomValue:101,102,104,105;SetValue:25</Leaf>
<Leaf ID="XL18" priority="1"
playerNode="false">CharInRoom:0,12%SetRandomValue:101,102,103,105;SetValue:26</Leaf>
<Leaf ID="XL19" priority="1"
playerNode="false">CharInRoom:0,9%SetRandomValue:101,102,103,104;SetValue:27</Leaf></Or>
<Leaf ID="XL13" priority="1"
playerNode="false">Value:14;Value:15;Value:16;Value:17;Value:18;Value:1001%</Leaf>
<Leaf ID="XL20" priority="3"
playerNode="false">%RemoveValue:1001;SetValue:1002;SetVisited:XL21,false</Leaf></And>
<Leaf ID="XL21" priority="0" playerNode="false">Value:1001%</Leaf></And><And ID="XA11" priority="1">
<Tree>Catalyst.xml</Tree>
<Leaf ID="XL22" priority="0" playerNode="false">Value:1002%</Leaf><Leaf ID="XL23" priority="1"
playerNode="false">DFrame:false%MoveCharacter:1,3300,1600,14;MoveCharacter:2,3273,1550,14;MoveCharacter:3,3360,166
6,14;MoveCharacter:4,3441,1666,14;MoveCharacter:5,3400,1620,14;Status:An explosion suddenly shakes the ship. It seems to
have come from the direction of the engine room.</Leaf>
<Leaf ID="XL24" priority="2"
playerNode="false">CharInRoom:0,14;CharInRoom:1,14;CharInRoom:2,14;CharInRoom:3,14;CharInRoom:4,14;CharInRoom:5,1
4;Or:Value:1||Value:2||Value:3||Value:4||Value:5%DialogueFrame:true,0,1,2,3,4,5</Leaf>
<Leaf ID="XL25" priority="4"
playerNode="false">DFrame:false%MoveCharacter:1,1660,2300,1;MoveCharacter:2,1670,2680,7;MoveCharacter:3,2690,1150,
11;MoveCharacter:4,1200,2825,0;MoveCharacter:5,3333,1600,14;RemoveValue:1002;SetValue:1003;SetVisited:XL22,false</Le
af>
</And>
<And ID="XA13" priority="1">
<And ID="XA14" priority="2">
<And ID="XA15" priority="1">
<Tree>Trent Post-Catalyst.xml</Tree>
<Leaf ID="XL27" priority="1"
playerNode="false">Value:1;Or:CharInRoom:1,1||CharInRoom:1,6%DialogueFrame:true,0,1</Leaf></And>
<And ID="XA16" priority="1">
<Tree>Jenny Post-Catalyst.xml</Tree>
<Leaf ID="XL28" priority="1"
playerNode="false">Value:2;CharInRoom:2,7%DialogueFrame:true,0,2</Leaf></And>
<And ID="XA17" priority="1">
<Tree>Percy Post-Catalyst.xml</Tree>
<Leaf ID="XL29" priority="1"
playerNode="false">Value:3;CharInRoom:3,11%DialogueFrame:true,0,3</Leaf>

```
                                        </And>
                                        <And ID="XA18" priority="1">
                                                <Tree>Rachel Post-Catalyst.xml</Tree>
                                                <Leaf ID="XL30" priority="1"
playerNode="false">Value:4;CharInRoom:4,0%DialogueFrame:true,0,4</Leaf></And>
                                        <And ID="XA19" priority="1">
                                                <Tree>Steve Post-Catalyst.xml</Tree>
                                                <Leaf ID="XL31" priority="1"
playerNode="false">Value:5;CharInRoom:5,14%DialogueFrame:true,0,5</Leaf></And><And ID="XA21" priority="1">
                                <And ID="XA22" priority="1"><And ID="XA23" priority="1">
                        <Tree>Trent Post-Weapons.xml</Tree>
                        <Leaf ID="XL44" priority="1" playerNode="false">Value:1%DialogueFrame:true,0,1</Leaf></And>
                                        <And ID="XA24" priority="1">
                                                <Tree>Jenny Post-Weapons.xml</Tree>
                                                <Leaf ID="XL45" priority="1"
playerNode="false">Value:2%DialogueFrame:true,0,2</Leaf></And>
                                        <And ID="XA25" priority="1">
                                                <Tree>Percy Post-Weapons.xml</Tree>
                                                <Leaf ID="XL46" priority="1"
playerNode="false">Value:3%DialogueFrame:true,0,3</Leaf></And>
                                        <And ID="XA26" priority="1">
                                                <Tree>Rachel Post-Weapons.xml</Tree>
                                                <Leaf ID="XL47" priority="1"
playerNode="false">Value:4%DialogueFrame:true,0,4</Leaf></And>
                                        <And ID="XA27" priority="1">
                                                <Tree>Steve Post-Weapons.xml</Tree>
                                                <Leaf ID="XL48" priority="1"
playerNode="false">Value:5%DialogueFrame:true,0,5</Leaf></And></And>

                                <Leaf ID="XL43" priority="0" playerNode="false">Value:1005%</Leaf></And></And>
                                <Or ID="XO2" priority="1"><Leaf ID="XL32" priority="1"
playerNode="false">Value:101%SetRandomValue:29,30,31,32</Leaf>
                                        <Leaf ID="XL33" priority="1"
playerNode="false">Value:102%SetRandomValue:28,30,31,32</Leaf>
                                        <Leaf ID="XL34" priority="1"
playerNode="false">Value:103%SetRandomValue:28,29,31,32</Leaf>
                                        <Leaf ID="XL35" priority="1"
playerNode="false">Value:104%SetRandomValue:28,29,30,32</Leaf>
                                        <Leaf ID="XL36" priority="1"
playerNode="false">Value:105%SetRandomValue:28,29,30,31</Leaf></Or>
                                <Leaf ID="XL26" priority="0" playerNode="false">Or:Value:1003||Value:1005%</Leaf>

                        <Leaf ID="XL38" priority="2" playerNode="false">CharInRoom:0,5;Value:91%Status:The keycard opens the
door. An alarm sounds, drawing the other passengers to the sealed
room.;RemoveValue:1003;SetValue:1004;SetVisited:XL26,false</Leaf></And>
                        <And ID="XA20" priority="1">
                                <Tree>Weapons Discovery.xml</Tree>
                                <Leaf ID="XL39" priority="0" playerNode="false">Value:1004%</Leaf>
                                <Leaf ID="XL40" priority="1"
playerNode="false">%Script:UnlockDoor;MoveCharacter:1,1600,1370,10;MoveCharacter:2,1520,1410,10;MoveCharacter:3,169
0,1310,10;MoveCharacter:4,1750,1250,10;MoveCharacter:5,1740,1180,10</Leaf>
                                <Leaf ID="XL41" priority="2"
playerNode="false">CharInRoom:0,10;CharInRoom:1,10;CharInRoom:2,10;CharInRoom:3,10;CharInRoom:4,10;CharInRoom:5,1
0;Or:Value:1||Value:2||Value:3||Value:4||Value:5%DialogueFrame:true,0,1,2,3,4,5</Leaf>
                                <Leaf ID="XL42" priority="4"
playerNode="false">DFrame:false%MoveCharacter:1,700,2550,6;MoveCharacter:2,1670,2680,7;MoveCharacter:4,1200,2825,0;
MoveCharacter:5,3333,1600,14;RemoveValue:1004;SetValue:1005;SetVisited:XL39,false;SetTimer:0</Leaf></And>
                        <And ID="XA28" priority="1">
```

```
                                <Tree>Death-Explosion.xml</Tree>
                                <Or ID="XO3" priority="2">
                                        <Leaf ID="XL50" priority="1"
playerNode="false">Not:CharInRoom:0,6%SetValue:81;PutCharacter:1,200,200,0</Leaf>
                                        <Leaf ID="XL51" priority="1"
playerNode="false">Not:CharInRoom:0,7%SetValue:82;PutCharacter:2,200,200,0</Leaf>
                                        <Leaf ID="XL52" priority="1"
playerNode="false">Not:CharInRoom:0,11%SetValue:83;PutCharacter:3,200,200,0</Leaf>
                                        <Leaf ID="XL53" priority="1"
playerNode="false">Not:CharInRoom:0,0%SetValue:84;PutCharacter:4,200,200,0</Leaf>
                                        <Leaf ID="XL54" priority="1"
playerNode="false">Not:CharInRoom:0,14%SetValue:85;PutCharacter:5,200,200,0</Leaf></Or>
                                <Or ID="XO4" priority="4">
                                        <Leaf ID="XL56" priority="1"
playerNode="false">Value:81%DialogueFrame:true,0,2,3,4,5</Leaf>
                                        <Leaf ID="XL57" priority="1"
playerNode="false">Value:82%DialogueFrame:true,0,1,3,4,5</Leaf>
                                        <Leaf ID="XL58" priority="1"
playerNode="false">Value:83%DialogueFrame:true,0,1,2,4,5</Leaf>
                                        <Leaf ID="XL59" priority="1"
playerNode="false">Value:84%DialogueFrame:true,0,1,2,3,5</Leaf>
                                        <Leaf ID="XL60" priority="1"
playerNode="false">Value:85%DialogueFrame:true,0,1,2,3,4</Leaf></Or>
                                <Leaf ID="XL49" priority="1"
playerNode="false">Value:1005;TimerExpired:0,60000;Not:CharInRoom:0,10;Not:CharInRoom:0,5;Not:CharInRoom:0,3;Not:Ch
arInRoom:0,4;Not:CharInRoom:0,2;Not:CharInRoom:0,1;DFrame:false%Status:An explosion sounds from the direction of the
weapons
room.;MoveCharacter:1,1600,1370,10;MoveCharacter:2,1520,1410,10;MoveCharacter:3,1690,1310,10;MoveCharacter:4,1750,
1250,10;MoveCharacter:5,1740,1180,10;RemoveValue:1005;SetValue:1006;SetVisited:XL26,false</Leaf>
                                <Leaf ID="XL55" priority="3"
playerNode="false">Or:Value:1||Value:2||Value:3||Value:4||Value:5;CharInRoom:0,10%</Leaf></And>

                        <Leaf ID="XL37" priority="1" playerNode="false">CharInRoom:0,5%Status:The door in front of you is sealed
shut.;SetValue:10</Leaf></And>
</And>


<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE And SYSTEM "forerunner.dtd">
<And ID="TA1" priority="2">


        <And ID="TA11" priority="1"><And ID="TA8" priority="7">
        <And ID="TA2" priority="1"><Leaf ID="TL8" priority="1" playerNode="true">%Dialogue:0,Who's Jenny?</Leaf>
                <Leaf ID="TL9" priority="2" playerNode="false">%Dialogue:1,My sister! Oh god, where is
she?;SetValue:8</Leaf></And><And ID="TA3" priority="1">
                <Leaf ID="TL10" priority="1" playerNode="true">%Dialogue:0,Jenny's all right.</Leaf><Leaf ID="TL11"
priority="2" playerNode="false">%Dialogue:1,You've seen her? Thank you! Thank you!</Leaf></And>



        </And><And ID="TA4" priority="9">
                <And ID="TA5" priority="1">
                        <And ID="TA6" priority="1"><Or ID="TO3" priority="3">
                                <And ID="TA7" priority="1">
                                        <Leaf ID="TL16" priority="1" playerNode="true">%Dialogue:0,I'll go see what I
can do.</Leaf>
```

<Leaf ID="TL17" priority="2" playerNode="false">%Dialogue:1,Go see what you can do. And hurry. We don't know how long we can last out here, or how long it will be before they can send help.</Leaf></And>

<And ID="TA9" priority="1">

<Leaf ID="TL18" priority="1" playerNode="true">%Dialogue:0,I can't get there.</Leaf>

<Leaf ID="TL19" priority="2" playerNode="false">%Dialogue:1,We'll need to find another way, then. That can't possibly be the only way on this entire ship of sending a message. Go see what you can find. I'll join you as soon as I'm feeling a bit more steady.</Leaf></And>

</Or><Leaf ID="TL14" priority="1" playerNode="true">%Dialogue:0,What should we do now?</Leaf><Leaf ID="TL15" priority="2" playerNode="false">%Dialogue:1,All right, we need to be calm. We need to appraise our situation. The crew obviously can't help us, so we'll need to figure things out on our own, see if there's a way to send a message for help without them. Can you get to the bridge?</Leaf></And>

<And ID="TA10" priority="1">

<Leaf ID="TL21" priority="1" playerNode="true">%Dialogue:0,Are you all right? You don’t look well.</Leaf>

<Leaf ID="TL22" priority="2" playerNode="false">%Dialogue:1,I twisted my ankle during the explosion. Hurts like hell, but I can’t focus on that right now. If you could find some meds, I sure would appreciate it. There’s a doctor around here.</Leaf></And>

</And>

</And><Or ID="TO1" priority="5">

<Leaf ID="TL5" priority="1" playerNode="true">%Dialogue:0,I can't find the crew anywhere.;SetValue:6</Leaf>

<Leaf ID="TL6" priority="1" playerNode="true">%Dialogue:0,There are other passengers in this section.;SetValue:7</Leaf>

</Or><Or ID="TO2" priority="8">

<Leaf ID="TL12" priority="1" playerNode="false">Value:6%Dialogue:1,Something may have happened to the crew. It's possible the only ones left alive are the passengers in this section of the ship.</Leaf>

<Leaf ID="TL20" priority="1" playerNode="false">Value:7%Dialogue:1,Someone should probably check on all of the other passengers to make sure they're all right.</Leaf></Or><Leaf ID="TL1" priority="1" playerNode="true">%Dialogue:0,I'm Neil.</Leaf><Leaf ID="TL2" priority="2" playerNode="false">%Dialogue:1,My name's Trent Porter. What happened here?</Leaf><Leaf ID="TL3" priority="3" playerNode="true">%Dialogue:0,Some kind of explosion. The ship's stopped.</Leaf><Leaf ID="TL4" priority="4" playerNode="false">%Dialogue:1,Where is everybody?</Leaf><Leaf ID="TL7" priority="6" playerNode="false">%Dialogue:1,Oh my god! Where's Jenny?</Leaf></And>

<Leaf ID="TL13" priority="1" playerNode="true">%Dialogue:0,I'll go look around.;SetValue:14;RemoveValue:1;SetVisited:XL2,false;SetVisited:TL13,false;DialogueFrame:false</Leaf></And>